4

ARL-FLIGHT-MECH-TM-412

AR-005-629

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Flight Mechanics Technical Memorandum 412

A DATA ACQUISITION PARALLEL BUS FOR WIND TUNNELS AT ARL (U)

by

J.F. Harvey

Approved for public release

AUGUST 1989

AR-005-629

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Flight Mechanics Technical Memorandum 412

A DATA ACQUISITION PARALLEL BUS FOR WIND TUNNELS AT ARL (U)

by

J.F. HARVEY

SUMMARY

A parallel bus is described for the interconnection of multiple intelligent slave microprocessor controlled peripherals to a master minicomputer. The bus enables the slaves to take snap shots of tunnel data with the exception of the multiplexed scanivalves, and then pass the data to the master on request.

DSTO ARL
MELBOURNE

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

CONTENTS

1. INTRODUCTION

The data acquisition system used by the Subsonic Wind Tunnel at Aeronautical Research Laboratory, Melbourne, was designed in 1972, and progressively commissioned over the following two years (Ref.1).

Tunnel productivity was further improved during 1982 with the installation of a dedicated computer (PDP11/44) which eliminated the time sharing access with the site's central computer. The dedicated computer also provides real-time processing of data and supports graphical displays for interactive operation with wind tunnel tests. ~~(Ref.2)~~.

These improvements, coupled with the requirement for faster and more accurate data measurements, have led to new instrumentation concepts based on microprocessor technology. These concepts are being implemented through replacement of the existing data acquisition system, which at March 1989 had registered more than 26214 hours of operation.

The data acquisition system used by the Transonic Wind Tunnel at Aeronautical Research Laboratory, Melbourne,was designed around a PDP8-I mini computer and has been in operation since mid 1967.

In recent years this dedicated minicomputer has become unreliable and the high cost of maintenance has led to its replacement with a microVAX which is intended to be connected to similar data gathering equipment to that used in the Subsonic Wind Tunnel.

A data bus was developed to provide bi-directional communication between the dedicated master mini computer and the numerous microprocessor slave modules ~~(Fig.1)~~ which form the new system. A[illegible] COMPUTE[illegible] [illegible]

2. GENERAL DESCRIPTION

2.1 Design Considerations.

The design requirements for the data bus are based on:

a) The bus must be faster than the dedicated master minicomputer.

b) Known and foreseeable bus applications for both wind tunnels involving real-time data acquisition, control and error handling.

c) The desirability to confine the main work load of the master to data processing, for peripheral outputs and for control command of the wind tunnel test program. Real-time local processing and servo control operations to be performed by the slave microprocessor modules within the distributed intelligent system.

d) The failure or removal of any slave module should leave the remainder of the system fully operational.

e) Many functionally different slave modules are to be connected to the instrumentation bus. Generally the modules would be permanently housed within the wind tunnel control room console; however, there may be exceptions such as the Strain Gauge Calibration Module which would be remotely located from the control room. Space, noise and other factors prevent the master mini computer being installed in the control room, thus necessitating bus lengths approaching 100 metres

2.2 Communication Considerations.

2.2.1 Master.

Since it is impractical for the main bus (Unibus of the PDP11/44 or Qbus of the microVAX) to extend outside the computer cabinet, a DR11-C card provides buffered interface access for external connections to user devices. Typically the DR11-C general device interface (Ref. 3) provides 16 bit parallel input/output data lines with handshake, all signals are TTL compatible. The majority of the signal lines have a 30 unit load capacity while the remainder are rated at 7 unit loads. Signal pulse widths in the order of 360 nanoseconds are involved and the allowable cable length is less than 8 metres.

Extending this bus using buffer stages was not a viable option for the wind tunnel system because of the limitations on drive capacity and cable length, along with the suspected problems associated with electrical interference on long single-ended TTL lines.

All the serial ports at the master were assigned to peripheral devices. For a new versatile wind tunnel system, serial ports were considered inappropriate for data transfer and control.

2.2.2 Serial.

Serial RS-232C communication is in present use between the master and the old 1972 instrumentation console of the Subsonic Wind Tunnel. Whilst serial communication is convenient and suitable for some applications, it was considered unsuitable for this application, as the number of slave devices is large, requiring up to 20 extra serial ports.

The upper speeds for standard serial communications using Reference 4 as a guide are:

Serial Ident.	Specified Drivers	No of Line Receivers	Type	Data Rate (Kbytes/sec)
RS-232	1	1	Single ended	2
RS-423-A	1	10	" "	30
RS-422-A	1	10	Differential	1000
RS-485	32	32	" "	1000

2.2.3. Parallel

In general, a parallel bus is faster and easy to implement; actual performance is governed by the choice of drivers and the width of the bus. Open collector drivers typically operate to a maximum of 0.25 megabits/sec., while Tri-state drivers may achieve the following data rates. (Ref.4).

Distance between Devices. (metres)	Data Rate. (megabits/sec.)
1	1
2	0.5

Considerable speed and noise immunity advantage are achieved with differential, rather than single ended line drivers/receivers. The comparison for half duplex operation on an 8 bit parallel bus is estimated as below.

Line Driver/Receiver Type	Data Rate (megabytes/sec.)
Single ended	0.5
Differential	10

The maximum data rates are intended only as a guide and are based on assumed drive, cable lengths and load conditions.

The well known IEE-488 bus, which is a master option, provides 8 bit parallel transfer with the upper data rate dependent upon the distance between communicating devices. However, an eight bit single ended system was not considered an optimum solution for this application.

Increasing the width of a parallel bus from 8 to 16 bits could be expected to double the data transfer rate measured in bytes/sec. However, program execution times for protocol, fetch and store operations need also to be considered and may dominate at the high rates.

2.2.4. Parallel Bidirectional Differential.

The decision was made to convert the single ended unidirectional 16 bit parallel data bus and control lines at the output of the DR11-C general device interface, to bidirectional differential data and control lines. Conversion is provided by a single, self-contained DR-11C Bus Interface (DBI) card (Fig.2).

A Bidirectional Parallel Interface (BPI) card (Fig.3) within each module, interfaces the Data Acquisition Parallel Bus (DAPB) to the slave micprocessor VME bus.

The DAPB is made from multi twisted-pair woven ribbon cable with regular parallel cable sections. Interconnection between the master and slaves is by press fit connectors onto the parallel cable sections. This allows slave modules to be readily positioned and connected to, or disconnected from, the master controlled network.

3. DR11-C/BUS INTERFACE (DBI).

3.1. Circuit Description.

A simplified signal flow diagram is shown in Figure 2,and the complete circuit diagram in Figure 10.

Latches combine the unidirectional 16 bit wide data buses DRIN and DROUT into a bidirectional 16 bit wide input/output bus (DAPB) using Tri-state RS-485 type driver/receivers. These driver/receivers provide the differential drive to remote slave modules during a master PASS (write) operation and accept differential input from slave modules during a master FETCH (read) operation. The two pulsed strobe lines, NDR (New Data Ready, 360 nanosecond positive TTL pulse) and DTX (Data Transmitted, a 360 nanosecond positive TTL pulse) are deglitched and along with CSR1 (a bit in the DR11-C Control Register, programmed to be an output) are functionally modified by the control logic. This control logic is programmed into two Programmable Array Logic (PAL) devices and is defined by the Boolean equations listed in Appendix 1 (Controlbus 1 and Deglitch 3).

The functionally modified control lines are buffered by Differential Bus Driver/Receivers, National Semiconductor type DS3695N. The "ORed" single ended input ERROR line is buffered and redesignated REQB (Request B line of the DR11-C interface).

3.2 Control Line Functions.

Control line directions and functions for the DR11-C interface card are:

INPUT LINES TO DBI FROM MASTER

NDR (New Data Ready)	360 nanosecond pulse.
DTX (Data Transmitted)	360 nanosecond pulse.
CSR0 (Control Status Register bit 0) CSR1 (Control Status Register bit 1)	Programmable Read/Write bits in the DR11-C registers.

OUTPUT LINES FROM DBI TO MASTER

REQA (Request A bit 7) REQB (Request B bit 15)	The state of bits 7&15 in the DR11-C, Control and Status Register is independently controlled by the DBI. The master program allows these bits to initiate an interrupt or be sensed as test flags within the master program.

INPUT LINES TO DBI FROM SLAVE

ACK	Acknowledge handshake,asserted by slave on acceptance of address and negated on data cycle.
ERROR	One or more slaves has encountered an error condition and an error code is available to the master.

OUTPUT LINES FROM DBI TO SLAVE

TRIG. (Trigger)	Starts synchronised read of data in all slave modules.
IFC (Interface Clear)	Initiates a system reset on all slave VME buses.
AS (Address Strobe)	Strobe to slave modules when output on differential bus is valid and represents an address .
DS (Data Strobe)	Strobe to slave modules when either data on bus is valid (Write) or when data from bus has been obtained (Read).

3.3 INTERFACE COMMUNICATION.

Most bus transactions are of the form:
Master writes address (16 bit word) then reads/writes data (16 bit word). The exceptions are:

a) IFC *where the master writes* address "FFFF" not followed by a read/write of data.

b) TRIG where the master sets the CSR0 bit high for a duration of approximately 20 microseconds.

c) ERROR where the slave detects an error condition and sets REQB high. REQB is program maskable and may initiate an interrupt in the master.

There is no Read/Write line within the interface. All even addresses are decoded as read (FETCH) where address bit 0 is low. A write (PASS) is decoded when address bit 0 is high, odd address.

3.3.1 Address format.

The address is of the form, 16 bit (Word) 2's complement binary, and has module specific address in the top byte (b8 to b15) with an interrupt vector number in the bottom byte (b0 to b7).

(b15.............b8) (b7.............b0)
(module specific address) (interrupt vector number).

The direction of the address is from master to slave only. The slave can never send an address to the master.

Bit 0 of the vector number is effectively the read/write line. Even interrupt vector numbers are decoded by the BPI as read vectors, while odd vector numbers are decoded as write vectors.

A bus read cycle has been designated a FETCH. Master reads data from a specific slave. A bus write cycle has been designated a PASS. Master writes data to a specific slave.It is not possible for the master to FETCH from more than one slave at any instant in time.

The top byte of the address is a module specific address and only activates one single slave module.

The slave modules cannot address the master, therefore it is not possible for two or more slaves to activate the bus at any given instant in time and hence cause bus contention and hang the bus.

All communication is initiated by the master; the slave can only flag an error condition as discussed in section 3.3.5.

Module specific addresses have been allocated as follows:

Development Module (Spare)	"84XX"
Strain Gauge Calibrator	"80XX"
Strain Gauge Sting Balance	"81XX"
Actuator/Hinge Moment	"82XX"
Scanivalves	"83XX"
Tunnel Air Velocity	"85XX"
Parameter Display	"87XX"
Sting Control	"88XX"
Auxiliary/Data	"89XX"
Inclinometer	"8AXX"
Unassigned	"86XX'

There is a possible range of 256 interrupt vectors within each slave module; however, each slave VME bus based module has need for its own internal interrupt vectors (such as an analogue conversion has been completed).Therefore vectors in the range Hex 0 to 5F have been designated as internal slave vectors and may not be addressed by the master.

The vector assignment varies from slave module to slave module but is usually of the following form.

VECTOR No.	FETCH/PASS	FUNCTION
60	F	Error code/data.
62	F	Slave Identification String
63	P	TRIG.
65	P	Clear status buffers.
67	P	Clear error flag.
Even Nos. 70-CE	F	Read data from slave.
Odd Nos. 71-CF	P	Write data to slave.
Even Nos. DO-DE	F	Read slave hardware status.
Odd Nos. F1-FF	P	Write slave hardware control.

3.3.2. Data Format.

The data is of the form 16 bit (word) offset binary. There are no restrictions on data written to slaves. However, data read from slaves should be in the range 0001 to FFFE.

The data words 0000 & FFFF are reserved,these are the extreme ranges of the data.

The majority of data are scaled in volts with the range +/-10.000 volts. The most significant bit (b15) is the sign bit. There are two zero conditions, binary 7FFF is +0.000 while 8000 is -0.000. One bit error is equivalent to 300 micro volts.

The read data word FFFF is reserved for future use while 0000 is designated a null.

3.3.3 IFC

The IFC is intended to be a system reset for all the VMEbus slave modules and is a common line to each. The master writes the address FFFF to the DROUT port of the DR11-C and asserts CSR1.The address FFFF is decoded in the DBI and the IFC line is asserted on the DAPB. The IFC should remain asserted for 200 milliseconds to enable all the MC68000 microprocessors to undergo a complete system reset. The master controls this duration by keeping CSR1 asserted for this time. Negating CSR1 removes the IFC.

3.3.4 TRIG.

The TRIG is common to all slave modules and once activated by the master asserting CSR0 of the DR11-C interface, initiates a simultaneous start of conversion of all analogue to digital converters and sampling of data from all external measuring devices. All slave modules respond, enabling a snap-shot of data to be taken of the wind tunnel conditions at that instant in time. There are no multiplexing devices other than scanivalves. All analogue input channels have individual analogue to digital convertors which finish conversion within 50 microseconds.

The TRIG should remain asserted for 20 microseconds and then the master negates CSR0, negating TRIG.

3.3.5 ERROR

This is the only way the slaves can obtain the attention of the master. All slaves are "ORed" onto this common DBI input.

Should a slave experience an error condition, the slave triggers the error latch in its individual BPI. The common bus error line is asserted causing a master interrupt.

The master then FETCH's the error code from each individual slave. Slaves not experiencing error conditions return a null (0000) while the slave or slaves experiencing the error return a code other than null when individually addressed. By initiating a second FETCH from that module returning an error code (from the same vector vec 60)a data word detailing the source of the error is returned. Multiple FETCH's from vec 60 may be required to fully determine the exact error source. Vec 60 should be multiple read until the null is returned, this indicates that all the error data has been obtained. The master writes a null to that slave module's clear error vector, ie Vec 67 (null). The error flag is then cleared in that slave and the master takes the appropriate action to process the error condition. If however, the error line is still asserted another slave is experiencing an error condition. The master interrogates the slaves again at Vec 60 to determine which slave and what is the problem.

3.4 DBI Timing.

3.4.1 Master Implements a FETCH Cycle (Fig.4)

a) Master asserts the flag CSR1 = 1
b) Master writes desired module address and even vector number into DROUT buffer.
c) DBI control logic responds to CSR1 = 1 and NDR pulsing high for 360 ns to latch the module address/vector onto the DAPB and asserts the address strobe (AS = 1).
d) Once the address is decoded by the BPI at the desired slave module the acknowledge handshake line ACK is asserted (ACK = 1).

e) The DBI negates the address strobe (AS=0) and asserts the data strobe (DS=1).
f) On the assertion of the data strobe (DS) the slave places the required data onto the DAPB and negates the acknowledge handshake line (ACK=0).
g) The DBI negates the data strobe (DS=0) and asserts REQA to signal the master to read the data from the DBI input data latch on port DRIN.
h) On the negation of the data strobe (DS=0) the slave releases the bus and completes its FETCH timing cycle.
i) The master reads the data from DRIN asserting DTX for 360 nanoseconds and allowing the DBI to negate REQA (REQA=0).
j) The master negates the flag CSR1=0 to complete the FETCH timing cycle.

3.4.2 Master implements a PASS cycle (Fig.5).

a) Master asserts the flags CSR1=1
b) Master writes desired module address and odd vector number into DROUT buffer.
c) DBI control logic responds to CSR=1 and NDR pulsing high for 360 ns. to latch the module address/vector onto the DAPB and asserts the address strobe AS=1.
d) Once the address is decoded by the BPI at the desired slave module the acknowledge handshake line ACK is asserted (ACK=1)
e) The DBI negates the address strobe (AS=0) and asserts REQA to signal the master to write the desired data into DROUT buffer.
f) On NDR pulsing high for 360 ns, the data on the DROUT buffer is latched onto the DAPB, the data strobe (DS) is asserted and REQA is negated and the master PASS cycle is completed.
g) On the assertion of the data strobe (DS) the slave reads the data, negates acknowledge handshake line ACK and completes its PASS cycle.
h) On the negation of the acknowledge handshake line ACK the DBI negates the data strobe (DS) and the PASS cycle is complete.

4. BIDIRECTIONAL PARALLEL INTERFACE (BPI)

This card resides on the VMEbus of each of the slave processing modules. It is an interface between the DAPB and the slave microprocessor.

4.1 VMEbus.

VMEbus (Ref.4) is a versatile 16/32 bit backplane bus approved by the IEEE P1014 Standard Committee. This bus has been specified both electrically and mechanically. The mechanical specification describes the physical dimensions of subracks, backplanes, front panels, plug-in boards etc. It has been designed to meet the Eurocard board and hardware format.

4.2 Circuit Description.

A simplified signal flow diagram is shown in Figure 3, and the complete circuit diagram in Figure 11. For the purpose of this text, asserted refers to the logic function becoming active and (*) following a signal name refers to the negation being the true state.

The DAPB input differential bus drivers/receivers are held in the receive state while the handshake acknowledge driver (ACK) is held Tri-stated.

Addresses appearing on the DAPB bus are decoded by a Module Decoder which is at a specific address as listed in section 3.3.1. A decoded address initiates the on board logic to activate the VMEbus interrupt (IRQ5). The slave microprocessor completes its current instruction and proceeds to service the interrupt. The vector associated with the least significant byte of the address is passed onto the VMEbus via Tri-state data buffers to identify the Master request. The least significant bit of the vector determines the read/write status of the following data On completion of the interrupt cycle the BPI flags the master by activating the handshake acknowledge line (ACK=1). Data is able to be written to the VMEbus or read from it depending upon whether it is a PASS or FETCH cycle. The data direction is determined and the DAPB bus drivers/receivers along with the VMEbus data buffers activated to handle the data transfer. The handshake acknowledge line is negated (ACK=0) and Tri-stated on the completion of the data cycle.

Whenever the TRIG is activated by the master the vector to the slave is determined by hard wired logic (Vec 63) and is not passed via the normal way. The VMEbus address and data bus remain inactive during this special cycle,the cycle being common to all slave processors.

Similarly the IFC activates the VMEbus system reset line and is again common to all slave processors.

An error latch within each slave BPI is set by the slave experiencing an error condition. The slave on setting the error latch also writes into string 1 buffer an error code followed by an error data word or words and terminated by a null. A typical error format is as follows:

```
     "VEC 60" -- "VEC 60" -- "VEC 60"
     "CD01"   -- "0005"   -- "0000"
     Code for --     |    --  null
 data overange       |
                  expand
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
 0   0   0   0   0   0   0  0  0  0  0  0  0  1  0  1
        "1" Overranged channel
        "0" Normal operation
        therefore channels 1 & 3 overranged.
```

The master on the first FETCH of "Vec 60" determines an error code or if an error does not exist within that slave a null is returned.

If an error code is returned the master does a second FETCH of "Vec 60" to determine the error data. As in the example above data of "0005" was returned; on expanding this binary word to bits, it can be determined that channels 1 and 3 are experiencing an overrange on their inputs.

The third FETCH of "Vec 60" returns a "null" signalling the end of the error string.

The error latch within the slave remains active after the error string is read and is reset by the master performing a PASS to "Vec 67" of data = null. The data must be sent to complete the PASS cycle.

Should the error line to the master remain activated then another slave module must also be experiencing an error condition; the above process is repeated until the error line returns to the inactive state.

The master may choose to ignore the error condition that is the prerogative of the master.

The Module Decoder (MD) is the only PAL to differ from module to module, the VMEbus Address Decoder (AD) PAL code remains the same as there is only one BPI card per slave module. The VMEbus interrupts are handled by an Interrupt Logic PAL (IL) while the Data Direction Logic is spread over the remaining two PAL's (DDL1 and DDL2). The Boolean equations for these PAL's are listed in Appendix 2.

4.3 BPI TIMING.

4.3.1 Master Implements a FETCH Cycle (Fig. 6)

The timing sequence may be divided into three sequential events.

Event 1: Master addresses a slave module and initiates an interrupt on the slave VMEbus.

Event 2: Slave VMEbus acknowledges the interrupt and initiates a VME interrupt cycle.

Event 3: Slave reads data from the VMEbus memory and places it on the DAPB for the master to read.

A more detailed description of these events is as follows:

Event 1: The DAPB becomes active; ie., the address strobe AS is asserted and a slave module address is placed on the bus. The slave module decoder asserts MODSEL* and the least significant bit of the address b0 is tested for read/write (AD0=0 is FETCH). The VMEbus interrupt (IRQ5*) is asserted and latch line VIA* asserted to inhibit further interrupts until the current cycle is completed.

Event 2: The slave microprocessor on completion of its current instruction cycle retains IACKIN* and proceeds to enter the interrupt routine pointed to by the vector (the lower byte of the master address). Once this action has taken place DTACKI* is returned onto the VMEbus.

Event 3: The master asserts the data strobe DS to initiate a VMEbus read cycle. When the slave responds by addressing the BPI and asserting VMESEL* (VMERW* remains negated), the data on the VMEbus is passed through to the DAPB. DTACKD* is asserted and the COMP* line activated to negate ACK on the DAPB and complete the FETCH cycle.

4.3.2 Master Implements a PASS Cycle (Fig.7)

This timing sequence is very similar to the FETCH cycle described above. The address b0 is odd (AD0=1 is Pass) which initiates a VMEbus write cycle.

4.3.3 Master Implements TRIG. (Fig.8)

TRIG being a common line to all slave modules has a simplified timing sequence common to all and may be divided into two sequential events.

Event 1: Common TRIG line is activated and all bus slave modules initiate an interrupt on their respective VMEbus.

Event 2: Each slave VMEbus acknowledges the interrupt and indicates a VME interrupt cycle.

Event 1: The TRIG line on the DAPB becomes active. The VMEbus interrupt (IRQ5*) is asserted and the latch line VIA* asserted to inhibit further interrupts until the current cycle completed.

Event 2: The slave microprocessor on completion of its current instruction cycle returns IACKIN* and proceeds to enter the interrupt routine pointed to by the hardwired TRIG Vector (VEC 63). Once this action has taken place DTACKI* is returned onto the VMEbus.

4.3.4 Master Implements IFC

The IFC is a common line to all slave modules. This line is buffered in the individual slave BPI cards and connected directly to the VMEbus SYSRESET* line.

IFC is asserted by the master and must remain asserted for 200 milliseconds to enable each slave microprocessor to undergo a complete system reset.

4.3.5 Slave Flags Master of Error Condition. (Fig.9)

The ERROR line is common to all slave modules and is an input to the master to signal a slave error condition. Each slave is connected in a wired "ORed" configuration.

Each BPI contains an error latch which is written to by the slave only. The VMEbus asserts the slave error address which is decoded to assert VMEATT* AND VMERW*, writing D0 of the VMEbus data into the latch. IF D0=1 the latch is set, if D0=0 the latch is cleared.

The master cannot clear the error latch, it can only ask the slave to do it.

DTACKD* is asserted to terminate the VMEbus write cycle.

5. MASTER SOFTWARE CONSIDERATIONS.

The master minicomputer in its control of the DR11-C General Device Interface should follow the procedures set out below for correct operation of the DAPB.

5.1 Fetch (Master <- Slave)

a) Assert CSR1 (CSR1 = 1).
b) Write Module Address/Vector to DROB port. DROB0=0for FETCH cycle.
c) Wait for REQA to be asserted (REQA = 1) - Indicating that the Module Address/Vector has been recognised by a slave. Time out if REQA not asserted after one millisecond of Writing Module Address/Vector to slave. No slave on DAPB to respond to that Module Address.
d) Negate CSR1 (CSR1 = 0).
e) REQA being asserted indicates that valid data is available on DAPB. Read data from DRIB port.
f) Wait for REQA to be negated (REQA = 0) - Indicating slave completed its write cycle. Time out if REQA not negated after one millisecond of reading slave data. Fault has occurred within slave, write cycle and data not valid.
g) FETCH cycle completed.

5.2 PASS (Master -> slave)

a) Assert CSR1 (CSR1 = 1).
b) Write Module Address/Vector to DROB port. DROB0 = 1 for PASS cycle.
c) Wait for REQA to be asserted (REQA = 1). Indicating that the Module Address/Vector has been recognised by a slave. Time out if REQA not asserted after one millisecond of Writing Module Address/Vector to slave. No slave on DAPB to respond to that Module Address.
d) Negate CSR1 (CSR1 = 0).

e) REQA being asserted indicates that the slave is ready to read data from DAPB. Write data to DROB port.

f) Wait for REQA to be negated (REQA = 0) - Indicating slave completed its read cycle. Time out if REQA not negated after one millisecond of writing data to slave. Fault has occurred within slave, read cycle incomplete and data not received.

g) PASS cycle completed.

5.3 TRIG (Master to all Slaves).

a) Assert CSR0 (CSR0 = 1).
b) Delay 20 microseconds.
c) Negate CSR0 (CSR0 = 0).
d) TRIG cycle completed.

5.4 IFC (Master to all Slaves).

a) Assert CSR1 (CSR1 = 1)
b) Write code "FFFF" to DROB port.
c) Delay 200 milliseconds.
d) Negate CSR1 (CSR1 = 0).
e) IFC cycle completed.

5.5 ERROR (All Slaves to Master)

Configure REQB to initiate Master interrupts.
a) REQB asserted (REQB = 1).
b) Address Vec 60 of all slaves to identify error (FETCH) cycle.
c) Address Vec 60 of module with error to obtain error data (FETCH cycle).
d) Continue to address Vec 60 of module with error until null detected (FETCH cycle).
e) Process error data.
f) Request slave with error to clear error latch. PASS Slave Module Address/Vec 67 with data = null. Vec 67 is a master request to clear the error latch.
g) If error still remains, ie REQB is still asserted, another slave module may have experienced an error. Repeat steps b to f above.

6. SLAVE SOFTWARE CONSIDERATIONS.

The slave MC68000 microprocessors operating on VMEbus should implement the following procedure for correct operation on the DAPB.

6.1 Vector Assignment.

The master when addressing a slave initiates an interrupt on IRQ5 of the slave VMEbus. Other levels of interrupt may also be active within the slave. For this reason the VMEbus user interrupt vectors have been divided into two groups.

Group 1: Vectors from vector 40 to vector 5F have been assigned to normal hardware interrupt vectors within the slave module. Hardware such as Analogue to Digital Convertors call on interrupt IRQ3 on end of conversion etc.

Group 2: Vectors from vector 60 to vector FF have been assigned to master FETCH/PASS functions on the slave module. The even vectors have been designated, read from slave memory or FETCH vectors. While the odd vectors have been designated, write to slave memory on PASS vectors.

This group of vectors has been further divided in the following way.

Even Vectors 60 - 6E FETCH Error/Data strings from slave.

Odd Vectors 61 - 6F PASS TRIG/Control of Buffers etc.

Even Vectors 70 - CE FETCH Read data words from slave.

Odd Vectors 71 - CF PASS Write data words to slave.

Even Vectors D0 - DE FETCH Read status buffers in slave.

Odd vectors D1 - DF PASS Unassigned.

Even Vectors FO - FE FETCH Read Special Functions.

Odd Vectors F1 - FF PASS Write Special Functions.

6.2 Error/Data Strings.

Error Strings are written to slave memory buffers with no restriction on the length of the string. The strings start with a code to identify the type of error, followed by data words containing either error data or ascii characters providing an error message, terminated by a null word.If the error code at the beginning of the string is missing and instead a null is read, there is no error string available from this slave module.

Data Strings are similiar except they do not start with a code but contain a string of data, either binary or ascii, again terminating with a null.

The Scanivalve module sends pressure port data in the form of data strings. There are six strings, one for each of the six Scanivalves. Scanivalves may have a maximum of forty eight words in each string, one word for each scanned port.

The Module Identification String contains an ascii train, two ascii characters per word. This String identifies the type of module and is read by the master as part of its initializing routine to identify powered modules on the DAPB for the current aerodynamic test.

6.3 Error Codes.

The most common error codes read by the master on receipt of an error interrupt from the slave are as follows:

Error Code	Meaning
AD01	Unassigned Vector - Read error
AD02	Unassigned Vector - Write error
ED00	System fault - Wrong vector
CD01	Opto Isolator Error - Amplifier Overange
BD01	ADC Conversion Time Out
BD02	ADC Data Overange

6.3.1 Unassigned Vectors - Read AD01/Write AD02

These error vectors are in the user interrupt vector table between vector 40 and vector 5F. They are reserved vectors for slave module hardware other than BPI.

If access to these vectors is attempted by the master these error codes are stored in the error string and the error line activated.

6.3.2 System Fault - Read/Write/Wrong Vector. ED00.

These error vectors are outside the user interrupt vector table between vector 0 and 5F. They are reserved vectors for slave microprocessor local use, such as Reset Vectors, Bus Error, Trap Instruction Vectors and Interrupt Autovectors. If access to these vectors is attempted by the master, the error code CD03 is stored in the error string and the error line activated.

The slave microprocessor returns to its initial start position and recommences operation.

7. PERFORMANCE

There are three factors which determine the speed of data transfer of the bus.

1. The speed of the Master computer and its associated DR11-C bus interface card in actual reading and writing to the bus interface.

2. The speed of the interface cards and the length of the parallel bus between (DBI/DAPB/BPI combination).

3.The speed of the Slave processor in reading and writing to the bus interface.

Some idea of the data transfer speed can be obtained from the Fetch and Pass timing sequences of figures 4 & 5.
The Master in asserting CSR1 and then providing the address (NDR) required 8.5 microseconds. Then once the data was placed on the bus and REQA asserted, the Master took another 520 microseconds to aquire the data. The Master required 528.5 microseconds of bus transfer time.

The speed of the bus interface cards and the parallel bus was extremely fast. Being TTL logic and Bipolar Gate Arrays the delay times measured over 100 metres of twisted paires bus cable was less than 500 nanoseconds.

The speed of the Slave 68000 microprocessor operating with a clock of 8 MHz was: To acknowledge the interrupt, 9 microseconds and to read/write data from bus ,16 microseconds. The Slave required 25 microseconds of bus transfer time.
The total bus transfer time is 554 microseconds, giving a data rate of 1800 16 bit data words per second. The Master (in this case micoVAX minicomputer) has the slowest response time requiring 95% of the transfer time. Excluding the master transfer time the bus/slave interface transfer data rate is in the order of 40 kilowords per second.

REFERENCES

1. Sear W.F.L.S., Sutton C.W. & Harvey J.F. — A Versatile Data Acquisition system for a Low Speed Wind Tunnel. ARL Aerodynamics Report 155 Nov.1980

2. Fairlie B.D. — A Real Time Data Acquisition System for a Low Speed Wind Tunnel ARL Aerodynamics Report 163, Feb.1985.

3. Digital Equipment — DR11-C General Corporation Device Interface User's Manual.

4. Texas Instruments 1985 — Linear & Interface Circuit Application Volume 2 Line Circuits Display Drivers

5. Motorola Rev. C.1 October 1985 — The VMEbus Specification

ACKNOWLEDGEMENTS

Of the many people who have made significant contributions to the development of this bus, the author would particularly wish to record the contributions of Messrs I.G.Powlesland and C.W.Sutton.

NOTE: MicroVAX, PDP11,PDP8,Unibus,Qbus and DR11-C are all registered trade marks of Digital Equipment Corporation.
Tri-State is a registered trade name of National Semiconductor Corporation.

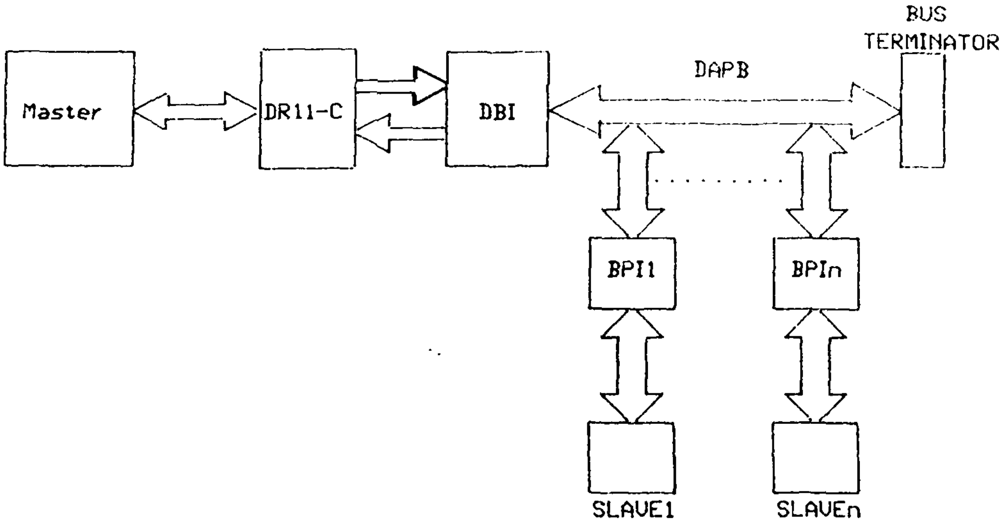


Fig.1 BUS CONFIGURATION

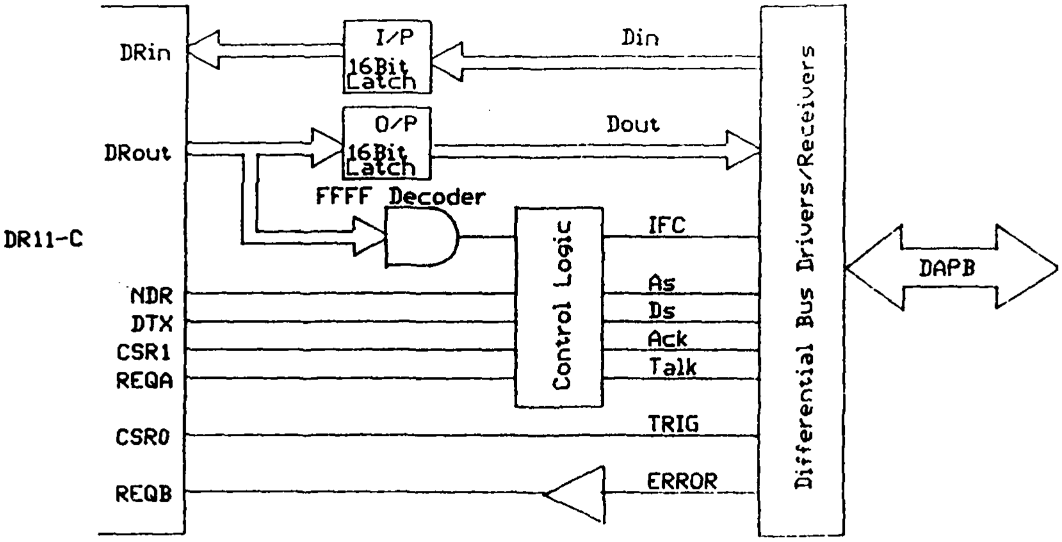


Fig.2 DBI Circuit Schematic

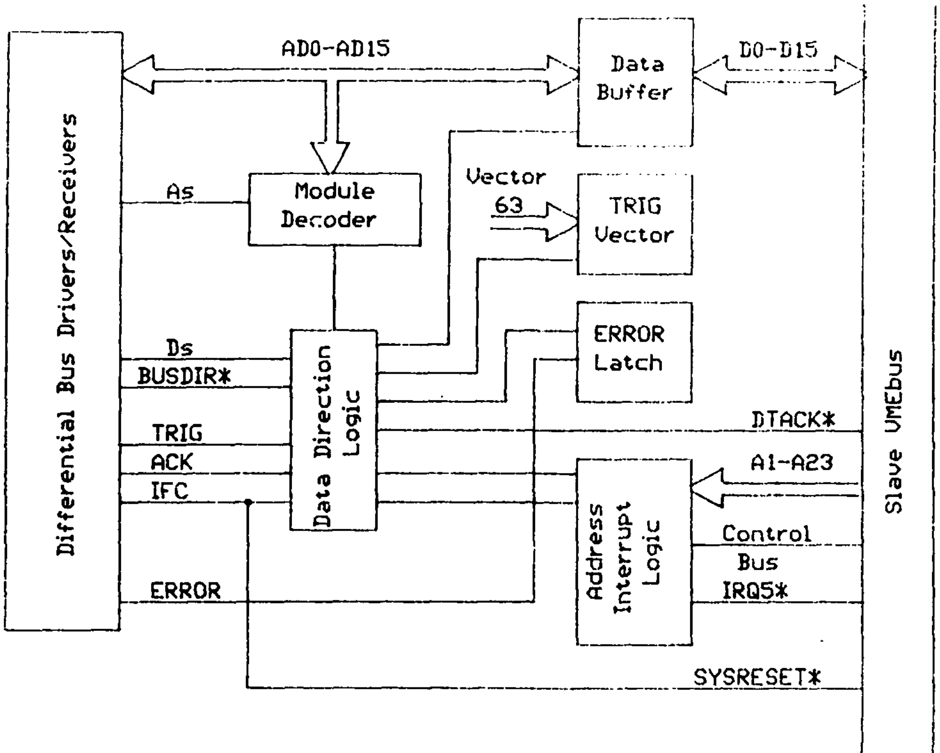
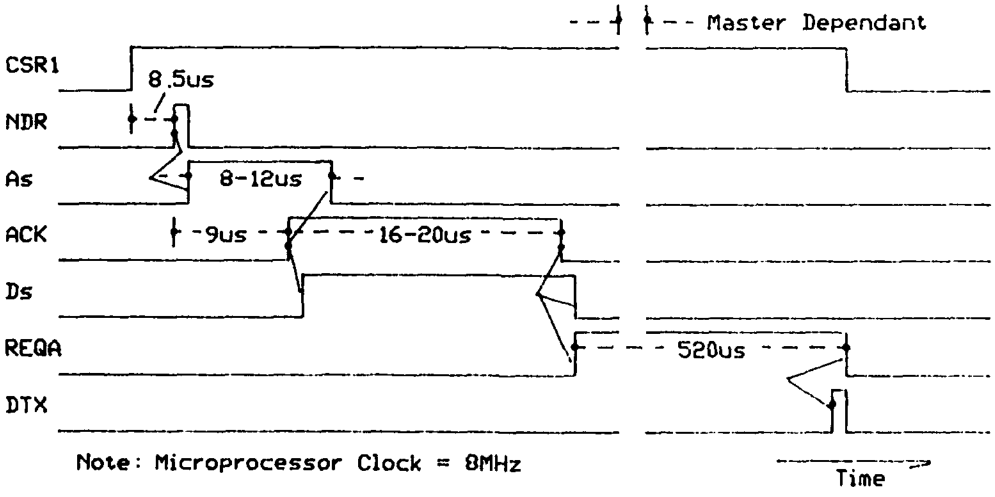


Fig.3 BPI Circuit Schematic



Note: Microprocessor Clock = 8MHz

Fig. 4 DBI Fetch Timing Sequence

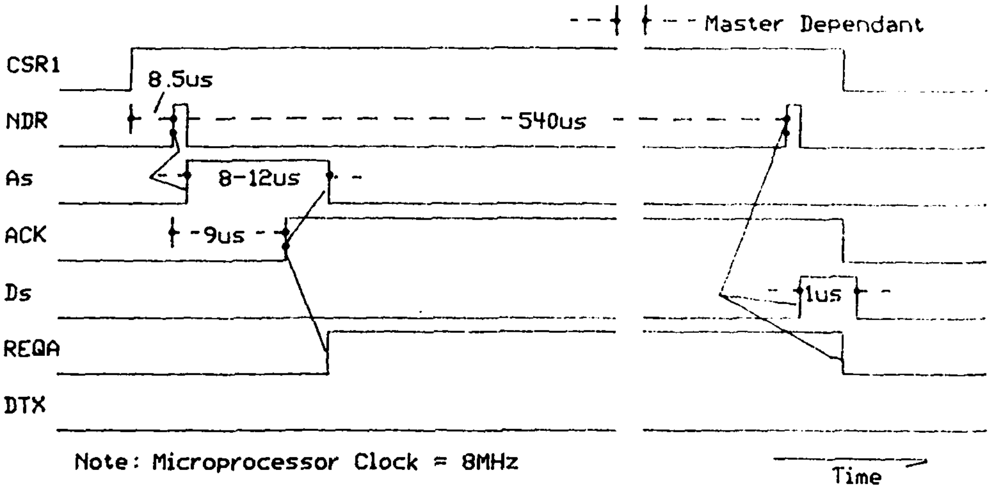


Fig.5 DBP Pass Timing Sequence

```
Master calls---|--VME bus-----|--VME bus write
                interrupt cycle       cycle
modsel*    11000  0000000000011  1111111111111111
ad0        11000  0000000000011  1111111111111111
irq5*      11100  0000011111111  1111111111111111
via*       11110  0000000000000  0000111111111111
selodd*    11111  1111111111111  1111111111111111
buswrite*  11111  1111111111111  1111111111111111
iackin*    11111  1000000000000  0111111111111111
intack*    11111  1100000111111  1111111111111111
ddelay*    11111  1110000011111  1111111111111111
dtacki*    11111  1111000001111  1111111111111111
ackx*      11111  1111111110000  0000011111111111
daten*     11111  1110000000111  1110000000000111
datdir*    11111  1110000000111  1111111111111111
ds         00000  0000000000000  0111111110000000
vmesel*    11111  1111111111111  1100000000001111
vmerw*     11111  1111111111111  1100000000001111
d1*        11111  1111111111111  1110000000111111
d2*        11111  1111111111111  1111000000011111
dtackd*    11111  1111111111111  1111111110000111
busdir*    11111  1111111111111  1100000001111111
comp*      11111  1111111111111  1111100000001111
```

Fig.6 BPI FETCH Timing Sequence

```
Master calls--|--VME bus-----|--VME bus read
               interrupt cycle    cycle

modsel* 11000  00000000000111  1111111111111111
ad0     XX111  11111111111XXX  XXXXXXXXXXXXXXXX
irq5*   11100  00001111111111  1111111111111111
via*    11110  00000000000000  0000000000111111
selodd* 11100  00000000000111  1111111111111111
buswrite*1110  00000000000000  0000000000111111
iackin* 11111  10000000000000  0011111111111111
intack* 11111  11000011111111  1111111111111111
ddelay* 11111  11100001111111  1111111111111111
dtacki* 11111  11110000111111  1111111111111111
ackx*   11111  11111111100000  0000000001111111
daten*  11111  11100000001111  1100000011111111
datdir* 11111  11100000001111  1100000011111111
ds      00000  00000000000000  0111111111111100
vmesel* 11111  11111111111111  1000000111111111
vmerw*  11111  11111111111111  1111111111111111
d1*     11111  11111111111111  1100000011111111
d2*     11111  11111111111111  1110000001111111
dtackd* 11111  11111111111111  1111000011111111
busdir* 11111  11111111111111  1111111111111111
comp*   11111  11111111111111  1111111110111111
```

Fig 7 BPI PASS Timing Sequence

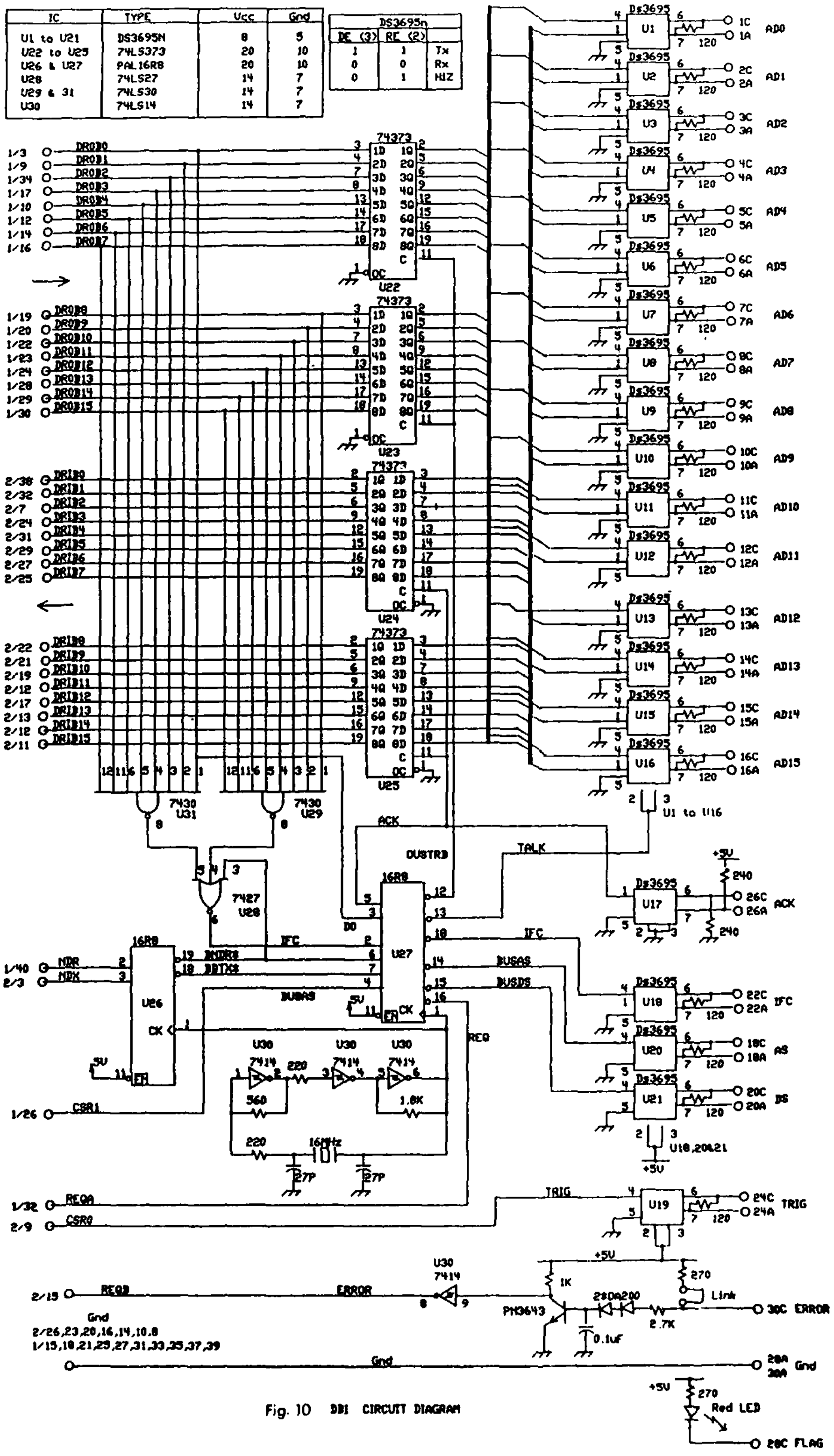
```
 Master calls---|---VME bus Interrupt cycle

trig  000011111  1111111111111111111111000000
irq5* 111110000  0000011111111111111111111111
via*  111111000  0000000000000000000000011111
iackin*11111111  1000000000000111111111111111
intack*11111111  1100000111111111111111111111
ddelay*11111111  1110000011111111111111111111
dtacki*11111111  1111000001111111111111111111
entv*  11111111  1110000000111111111111111111
```

Fig.8 BPI TRIG Timing Sequence

```
vmeatt* 11110000011111
vmerw*  11110000011111
d1*     11111000001111
d2*     11111100000111
dtackd* 11111110001111
efclkx* 11111111001111
```

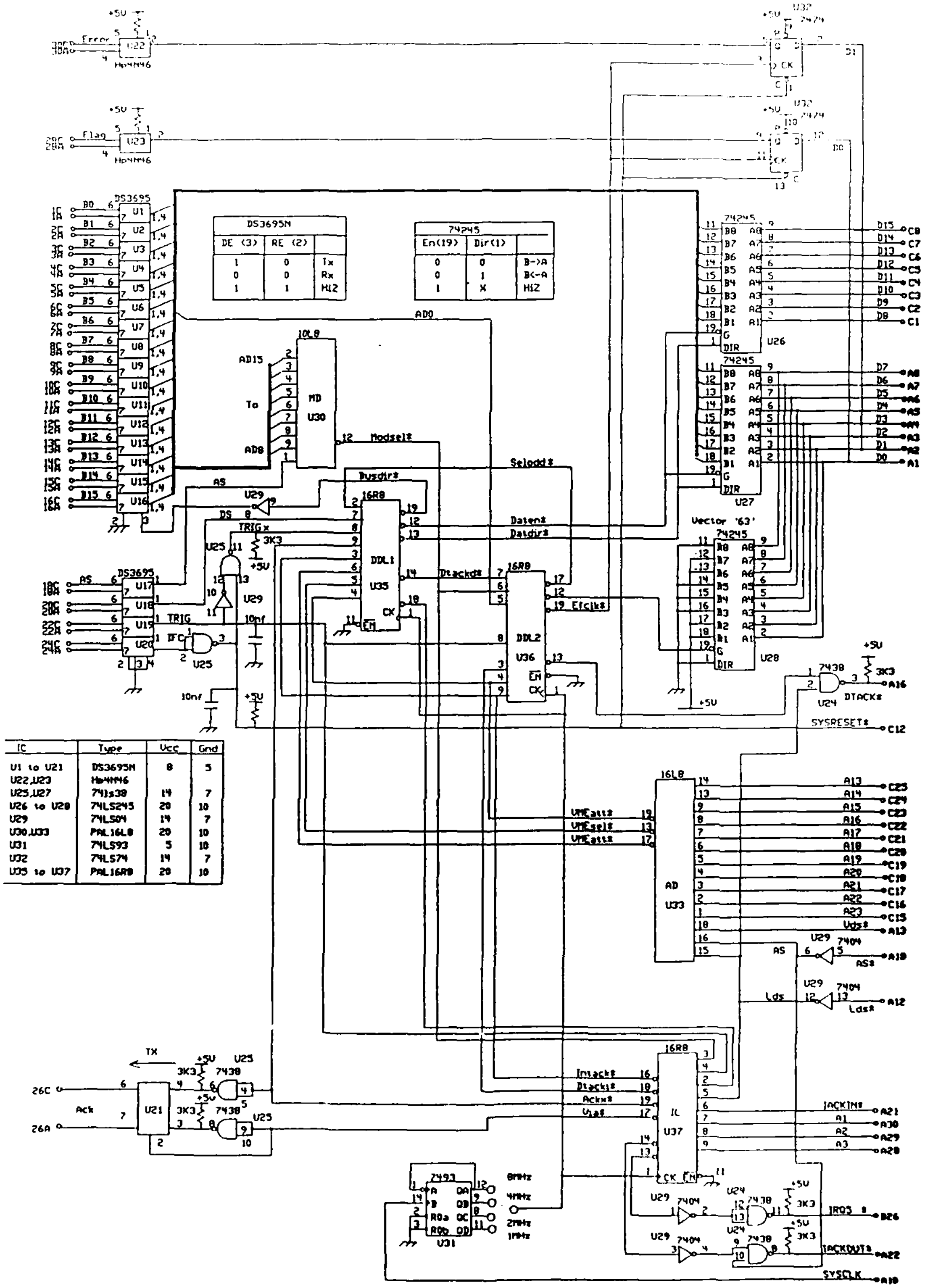
Fig.9 BPI ERROR Timing Sequence



IC	TYPE	Vcc	Gnd
U1 to U21	DS3695N	8	5
U22 to U25	74LS373	20	10
U26 & U27	PAL16R8	20	10
U28	74LS27	14	7
U29 & 31	74LS30	14	7
U30	74LS14	14	7

DS3695n		
DE (3)	RE (2)	
1	1	Tx
0	0	Rx
0	1	HIZ

Fig. 10 DBI CIRCUIT DIAGRAM



DS3695N		
DE (3)	RE (2)	
1	0	Tx
0	0	Rx
1	1	HIZ

74245		
En(19)	Dir(1)	
0	0	B->A
0	1	B<-A
1	X	HIZ

IC	Type	Vcc	Gnd
U1 to U21	DS3695N	8	5
U22,U23	Hp4N46		
U25,U27	74ls38	14	7
U26 to U28	74LS245	20	10
U29	74LS04	14	7
U30,U33	PAL16L8	20	10
U31	74LS93	5	10
U32	74LS74	14	7
U35 to U37	PAL16R8	20	10

Fig. 11 RPI CIRCUIT DIAGRAM

APPENDIX 1

PAL Boolean Equation for Controlbus1
PAL type PAL16R8 -- Control signals for correct bus operation.

```
Pin assignment:
clk ifc    d15  as44  ack   /bndr /bdtx i1   i2  gnd
/oe oustrb talk busas busds req   read  ifco op7 vcc

PAL Eqations:
/oustrb  := /bndr + ifc
/talk    := read * ack * /ifc + read * ack * /as44
/busas   := /as44 * //busas + /oustrb * /busas + ack + bndr + bdtx
              + ifc * as44
/busds   := read * /ack + /busds * /read * /oustrb + /ack + bdtx
              + ifc * as44
/req     := /read * /ack + ifc * as44 + read * ack + read * busas
              + read * oustrb + bdtx
/read    := bdtx + bndr * d15 + bndr * /as44 + /read * d15
              + /read * /as44 + /read * /bndr
/ifco    := /ifc * /ifco + /ifc * /as44
/op7     := ack
```

PAL Boolean Equations for deglitch3
PAL Type PAL16R8 -- Deglitcher for DR11-C output signals.
Ensures a single pulse of one clock period is produced once three consecutive samples of the input have been found to be asserted.

```
Pin Assignments:
clk ndrin dtxin in1   in2  in3  in4   in5  in6  gnd
/oe dtx1  dtx2  dtxlo nrd1 nrd2 ndrlo bdtx bndr vcc

PAL Equations:
/dtx1    := /dtxin
/dtx2    := /dtx1
/dtxlo   := /dtxlo * /dtx1 + /dtxlo * /dtx1 + /dtxlo * /dtx2
              + /dtxin * /dtx1 * /dtx2

/bdtx    := /dtxin * dtx1 * dtx2 * /dtxlo

/ndr1    := /ndrin
/ndr2    := /ndr1
/ndrlo   := /ndrlo * /ndrin + /ndrlo * /ndr1 + /ndrlo * /ndr2
              + /ndrin * /ndr1 * //ndr2

/bndr    := ndrin * ndr1 * ndr2 * /ndrlo
```

APPENDIX 2

```
PAL Boolean Equations for Module Decoder (MD)
PAL Type  PAL10L8   -  Decodes the module specific bus address

Pin Assignments:
AS AD15   AD14 AD13 AD12 AD11 AD10 AD9 AD8 gnd
nc Modsel nc   nc   nc   nc   nc   nc  nc  vcc

PAL Equations:
/Modsel = AD15 * /AD14 * /AD13 * /AD12 * Ad11 * /AD10 * /AD9 * /AD8
          * AS
; Module Address Hex '8700'  -  Parameter Display Module
```

```
PAL Boolean Equations for VMEbus Address Decoder (AD)
PAL Type  PAL16L8   -  Decodes the VMEbus specific Bus address'es

Pin Assignments:
A23   A22    A21 A20 A19 A18 A17   A16 A15    gnd
Write VMEsel A14 A13 Lds As  VMErw Uds VMEatt vcc


PAL Equations:
IF (vcc) /VMEsel = A23 * A22 * A21 * A20 * A19 * A18 * A17 * A16
                   * A15 * A14 * /A13 * As * Lds * /Uds
                    ; Address Hex 'FFCXXX' -word read/write
IF (vcc) /VMEatt = A23 * A22 * A21 * /A20 * A19 * A18 * A17 * A16
                   * A15 * A14 * /A13 * As * Lds * /Write
                    ; Address Hex 'EFCXXX' -byte write
IF (vcc) /VMErw  = As * Lds * /Write + As * /Uds * /Write
                    ; Lower Byte Write OR Upper Byte Write
```

PAL Boolean Equations for Interupt Logic (IL)
PAL Type PAL16R8 - VMEbus Interrup Handler

```
Pin Assignment:
Clk Comp   Modsel Trig    Lds Iackin A1  A2     A3   gnd
En  Ddelay Irq5   Iackout nc  Intack Via Dtacki Ackx vcc

PAL Equations:
/Irq5    := /Modsel * Via * /Trig + Trig * Via + /Irq5 * Dtacki
             ; One only Interrupt to call, lock out through Via.
/Via     := /Irq5 + /Via * Comp * /Trig + /Via * Trig
             ; Lock out until /Comp OR end of Trig.
/Intack  := /Irq5 * A3 * /A2 * A1 * Lds * /Iackin * /Via
             ; Interrupt acknowledge level 5.
/Ddelay  := /Intack      ; Delay one clock pulse.
/Dtacki  := /Ddelay      ; Release at end of VME strobe.
/Ackx    := Ddelay * /Dtacki * /Trig + /Ackx * Comp
             ; Latch untill /Comp.
/Iackout := /Iackin * Inlock       ; No daisy chain if this
interrupt is active.
/Inlock  := /Irq5 * A3 * /A2 * A1 * Lds  ; Is interrupt for this
card
```

PAL Boolean Equations for Data Direction Logic (DDL1)
PAL Type PAL16R8 - Logic to control the direction of data transfers, Trig vectors and Error/Flag latch.

```
Pin Assignments:
Clk Selodd Intack VMEatt VMEsel   VMErw Ds Trig Ackx   gnd
En  Datdir Daten  Dtackd Buswrite D1    D2 Comp Busdir vcc

PAL Equations:
/Buswrite  := /Selodd + /Buswrite * Comp
               ; Odd vector - Write to VMEbus.
/Daten     := /Intack * /Trig + Ds * /Vmesel * /Trig
               + /Daten * /VMEsel
               ; Data buffer enable strobe.
/Datdir    := /Intack * /Trig + Ds * /VMEsel * /Buswrite * /VMErw
                * /Trig + /Datdir * Ackx
               ; Data buffer direction control.
/D1        := /VMEsel * Ds + /VMEatt * /VMErw  ; Delay term
/D2        := /D1          ; Delay term
/Dtackd    := /Buswrite * /D2 * /VMEsel + Buswrite * /D2 * /VMEsel
               * /Ds + /D2 * /VMEatt + /Dtackd * /VMEsel
               ; Return Dtack for VMEbus read/write cycle.
/Comp      := /Buswrite * VMEsel * /D2 + Buswrite * /D2 + Trig
               ; Completion of data transfer flag.
/Busdir    := Ds * /VMEsel * Buswrite * /VMErw * /Trig
               + /Busdir * Ds
```

```
PAL Boolean Equations for Data Direction Logic (DDL2)
PAL Type PAL16R8   - To much to fit into one PAL this PAL contains
the overflow.

Pin Assignment:
Clk nc   Dtacki VMEatt A0 Modsel Dtackd Trig   Intack gnd
En  Entv Dtack  nc     nc nc     Selodd Efclkx Efclk  vcc

PAL Equations:
/Entv    := Trig * /Intack + Trig * /Dtacki   ;Enable strobe for
Trig vector.
/Efclkx  := /Vmeatt * /Dtackd     ; Clock for Error/Flag latch.
/Efclk   := Efclkx

/Dtack   := Dtacki * Dtackd       ; Dtack is Dtacki OR Dtackd

/Selodd  := /Modsel * A0          ; Module select with odd address
```

```
Note:   The symbol * represents the AND function.
        The symbol + represents the OR function.
        The symbol / represents the Inversion.
        The symbol := represents action taken on next Clock pulse.
```

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central

Chief Defence Scientist
FAS, Science Corporate Management (shared copy)
FAS Science Policy (shared copy)
Director, Departmental Publications
Counsellor, Defence Science, London (Doc Data Sheet Only)
Counsellor, Defence Science, Washington (Doc Data Sheet Only)
SA to the Thailand Military R and D Centre (Doc Data sheet only)
SA to the D R C (Kuala Lumpur) (Doc Data sheet only)
OIC TRS, Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)
Defence Industry and Materiel Policy, FAS

Aeronautical Research Laboratory

Director
Library
Chief - Flight Mechanics and Propulsion Division
Head - Flight Mechanics Branch
Branch File - Flight Mechanics
Author: J.F. Harvey (3 copies)
N. Pollock
N. Matheson (2 copies)
C.W. Sutton

Materials Research Laboratory

Director/Library

Defence Science & Technology Organisation - Salisbury

Library

Navy Office

Navy Scientific Adviser (3 copies Doc Data sheet)
Aircraft Maintenance and Flight Trials Unit

Army Office

Scientific Adviser - Army (Doc Data sheet only)
Engineering Development Establishment Library

Air Force Office

Air Force Scientific Adviser (Doc Data sheet only)
Aircraft Research and Development Unit
Scientific Flight Group
Library
Engineering Division Library

Statutory and State Authorities and Industry

Aero-Space Technologies of Australia, P/L Manager/Librarian (2 copies)

Universities and Colleges

Adelaide
Barr Smith Library

Flinders
Library

LaTrobe
Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library

New England
Library

Sydney
Engineering Library

NSW
Physical Sciences Library
Library, Australian Defence Force Academy

Queensland
Library

Tasmania
Engineering Library

Western Australia
Library
RMIT
Library

University College of the Northern Territory
Library

SPARES (10 copies)

TOTAL (73 copies)

AL 149

DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER AR-005-629	1b. ESTABLISHMENT NUMBER ARL-FLIGHT-MECH-TM-412	2. DOCUMENT DATE AUGUST 1989	3. TASK NUMBER DST 86-057

4. TITLE
A DATA ACQUISITION
PARALLEL BUS FOR WIND TUNNELS
AT ARL

5. SECURITY CLASSIFICATION
(PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF. (C) RESTRICTED (R), UNCLASSIFIED (U)).

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES
28

7. NO. REFS.
5

8. AUTHOR(S)
J.F. HARVEY

9. DOWNGRADING/DELIMITING INSTRUCTIONS
Not applicable

10. CORPORATE AUTHOR AND ADDRESS
AERONAUTICAL RESEARCH LABORATORY
P.O. BOX 4331, MELBOURNE VIC 3001

11. OFFICE/POSITION RESPONSIBLE FOR:
SPONSOR DSTO
SECURITY -
DOWNGRADING -
APPROVAL CFPD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH ASIDS, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO . . . .
No limitations

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE ☒ UNRESTRICTED OR ☐ AS FOR 13a.

14. DESCRIPTORS
Wind tunnels
Data acquisition

15. DRDA SUBJECT CATEGORIES
0051f

16. ABSTRACT

A parallel bus is described for the interconnection of multiple intelligent slave microprocessor controlled peripherals to a master minicomputer. The bus enables the slaves to take snap shots of tunnel conditions within a 50 microsecond period of all tunnel data with the exception of the multiplexed scanivalves, and then pass the data to the master on request.

PAGE CLASSIFICATION UNCLASSIFIED	
PRIVACY MARKING	

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT).

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER FLIGHT MECHANICS TECHNICAL MEMORANDUM 412	19. COST CODE 556411	20. TYPE OR REPORT AND PERIOD COVERED

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)